Question 1 – Studying Smart

Ali is a student who likes to procrastinate his work. He has an exam for a subject today and he decided to do his revision *N* minutes before his exam. The subject covers several topics. Ali knows very well for each topic how much marks it contributes and how much time it takes him to study. Assuming that he cannot choose to study any topic partially (i.e. either completely master the topic, or completely discard it), he wants to know the maximum marks that he can attain if he studies the topics optimally within *N* minutes.
Your task is to help Ali find the maximum marks that he could attain by studying the topics optimally within *N* minutes.

Input Format

The first line contains two space-separated integers, Z and N. Z denotes the number of topics in the subject and N denotes the remaining minutes before the exam. The subsequent Z lines contains two space-separated integers Ri and Ti. Ri denotes the marks contributed by the i-th topic in the exam, whereas Ti denotes the time taken (in minutes) to revise the i-th topic.

Constraints

- **The number of chapters in the subject, Z**: A positive integer where (1 <= *Z* <= 30)
- **The time remaining for exam, N in minutes**: A positive integer where (1 <= *N* <= 150)
- **The marks contributed by i-th topic, Ri**: A positive integer where (1 <= *Ri* <= 25)
- **The time taken to revise i-th topic, Ti in minutes**: A positive integer where (1 <= *Ti* <= 25)

Output Format

Output the maximum total mark and maximum time, separated by a space.

Sample Input 0

```
5 30
10 4
16 5
22 10
29 17
23 13
```

Sample Output 0

```
61 28
```

Explanation 0

Ali could attain a maximum mark of 61 (16 + 22 + 23) by studying chapters 2,3, and 5 within 28 (5 + 10 + 13) minutes.

Question 2 – Vocabulary Game

William is developing an educational game for kindergarten children to help them improve their English vocabulary. In this game, a word called *reference word* is given. The player's goal is to construct as many English words as possible using the letters from the *reference word*. The game's development is almost complete except for a routine to check the validity of all the words constructed by the player. A word is valid if all of its letters can be taken from the *reference word*, regardless of the letter case. For example, if the *reference word* is "prOgraMmIng", then some of the valid words are "roam", "groan", and "organ". The word "pop" is invalid as it contains an extra letter 'p' that is not available in "prOgraMmIng".
Given the *reference word*, and a list of English words constructed by the player, your goal is to determine for each player word whether it is valid or invalid.

Input Format

The first line contains the reference word. The next line contains a positive integer, M indicating the number of player words. This line is followed by M lines, which contains one player word each.

Refer the sample input for illustration.

Constraints

- **The reference word:** A string of alphabets (A-Z and a-z).
- **The number of player words, M:** A positive integer where (1<= M <=100).
- **The player word(s):** A string of alphabets (A-Z and a-z).

Output Format

- The output begins with the reference word single line.
- This line is followed by M lines where the i-th line (1 <= i <= M) contains the i-th player word, a colon, and a string (“Valid” if the i-th player word is valid or “Invalid” if the i-th player word is invalid, all without the double quotes.)

Refer to the sample output for illustration.

Sample Input 0

```
tRANSPorTaTIOn
5
rotation
script
position
snare
```

```
artist
```

Sample Output 0

```
tRANSPorTaTIOn
rotation:Valid
script:Invalid
position:Invalid
snare:Invalid
artist:Valid
```

Explanation 0

Each line shows the validity of the player word - whether it could be constructed from the reference letter.

Question 3 – Integer Expression

An integer expression is a mathematical expression that consists of integer operands and basic arithmetic operators (addition(+), subtraction(-), multiplication(*), and division(/)).

In this problem, you are to write a program that evaluates an integer expression and outputs the result of the expression.

Input Format

The first line contains the integer expression, which consists of operands and operators.

You may assume that the input integer expression will not encounter division by zero during the evaluation of the expression. You may also assume that the number of operands, N in the integer expression will be at least 2 and not more than 50.

Refer the sample input for illustration.

Constraints

- **The operand R:** A positive integer where (1 < R < 1000).
- **The operator:** An operator symbol (addition(+), subtraction(-), multiplication(*), and division(/)).

Output Format

Output the result of the integer expression after evaluation. The result should be an integer. Note that the rules of operator precedence holds.

Refer the sample output for illustration.

Sample Input 0

```
100-20*3+9/18-10
```

Sample Output 0

```
30
```

Explanation 0

The output is the result of the evaluation of the integer expression.

Question 4 – Danny and Sticks

Danny is a toddler whose hobby is to collect sticks. He has *N* number of sticks and upon observing, Danny sees that some of his sticks are of the same length. He is curious whether he could make squares out of those sticks and wants to know how many squares and of what size he could make out of those sticks.

Since there are many sticks at hand, help Danny to calculate the maximum area of the biggest square that can be formed and how many such squares can be made using the sticks.

Input Format

The first line contains a positive integer denoting the number of sticks. The next line contains N space-separated integers, each denotes Ri the length of i-th stick.

Refer the sample input for illustration.

Constraints

- **The number of sticks, N**: A positive integer where (1 <= *N* <= 1000).
- **The length of i-th stick, Ri**: A positive integer where (1 <= *Ri* <= 50)

Output Format

Output an integer denoting the maximum area of the square that could be formed, followed by a space and a positive integer denoting the number of such squares.

Refer the sample output for illustration.

Sample Input 0

```
7
5 3 2 3 6 3 3
```

Sample Output 0

```
9 1
```

Explanation 0

The area of the largest formable square is 9 (3 x 3) and only 1 such square can be formed from the sticks available.

Question 5 – A Thrifty Vacation

Siva is planning to travel to *N* number of cities in South India during the upcoming festival holiday. However, as he is a complete newbie to the location, he hires a local tour guide to assist his travel plan. The tour guide provides Siva with a list of travel cost to and from between each pair of cities. As Siva is a thrifty person, he wants to spend as less as possible while travelling to all the cities. Your task is to help Siva to find the most cost-efficient travel route that would enable him to visit every *N* number of cities exactly once and return to the origin city.

Input Format

The first line contains the number of cities, *N*. The following *N* lines contains *N* integers, denoting the adjacency matrix, which represents the cost of travel between each pair of cities.

Refer the sample input for illustration.

Constraints

- **The number of cities, N:** A positive integer where (3 <= N <= 50).
- **The cost of travel between a pair of city, C :** A positive integer where (1 <= C <= 1000)

Output Format

Output the the least costing path's city indices in sequential order.

Refer the sample output for illustration.

Sample Input 0

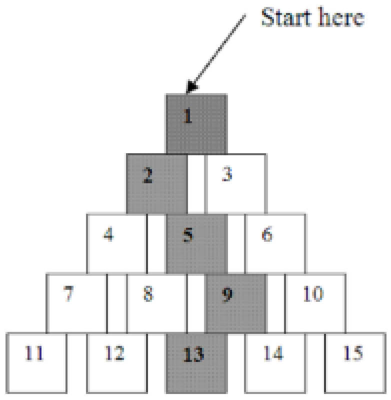
```
9
000 374 200 223 108 178 252 285 240 356
374 000 255 166 433 199 135 095 136 017
200 255 000 128 277 128 180 160 131 247
223 166 128 000 430 047 052 084 040 155
108 433 277 430 000 453 478 344 389 423
178 199 128 047 453 000 091 110 064 181
252 135 180 052 478 091 000 114 083 117
285 095 160 084 344 110 114 000 047 078
240 136 131 040 389 064 083 047 000 118
356 017 247 155 423 181 117 078 118 000
```

Sample Output 0

```
1 5 3 2 9 7 4 6 8
```

Question 6 – Pyramid Adventurer

A 2-Dimensional pyramid has a number of chambers as illustrated in the figure below. As an adventurer, you are required to work your way from the pinnacle of the pyramid to its bottom. Indicated inside a chamber is a number representing the reward that will be collected if you travel through that chamber on your way to the bottom. From each chamber, you are only allowed to move downwards to the chamber diagonally on the left or right. For example, from the chamber with reward 5, you can move to either chamber with reward 8 or 9. However, you cannot go to chamber with reward 7 or 10 from the one with reward 5. If you take the path as shown by the sequence of chambers shaded gray, the total reward will be 1+2+5+9+13 = 30.



Your task is to write a program that computes the number of paths contained within a given pyramid whose reward is equals a given number *R*.

Input Format

The input begins with a line containing two integers *N* and *R*. *N* represents the height of the pyramid, that is the number of rows in the pyramid and *R* represents the reward. The next N lines represents the chamber matrix which contains the reward values for each chambers in the row followed by a number of zeros to accomodate the rest of the chamber matrix.

Refer to the sample input for illustration.

Constraints

- **The height of the pyramid N**: A positive integer where (4 <= *N* <= 10).
- **The reward R**: A positive integer where (1 <= R <= 100).
- **The reward value of the chamber V**: A positive integer where (0 <= V <= 20).

Output Format

Output the number of paths whose reward equals to *R*.

Sample Input 0

```
5 29
1 0 0 0 0
2 3 0 0 0
4 5 6 0 0
7 8 9 10 0
11 12 13 14 15
```

Sample Output 0

```
2
```

Explanation 0

There are 2 paths in the pyramid whose reward equals to 29.

Question 7 – Nearest Pair

The nearest pair problem is to find two points that are nearest to each other in a given set of points. This problem is prevalent in numerous applications, one of which is in air-traffic control where you may want to monitor planes that come too close together, since this may indicate a possible collision.

Your task is to write a program that calculates the shortest distance that could be formed from a set of points.

Input Format

The first line contains a positive integer, N which represents the number of points.This line is followed by N lines which contains two integers, X and Y. X represents the x-coordinate of the point and Y represents the y-coordinate of the point.

Refer the sample input for illustration.

Constraints

- **The number of points, N:** A positive integer where (0 < N < 10000).
- **The x-coordinate, X:** A real number where (-10000.0 <= X <= 10000.0).
- **The y-coordinate, Y:** A real number where (-10000.0 <= Y <= 10000.0).

Output Format

Output the shortest formable distance from the set of points to a precision rounded off to 6 decimal places.

Refer the sample output for illustration.

Sample Input 0

```
8
-1 3
-1 -1
1 1
 2 0.5
 2 -1
 3 3
 4 2
 4 -0.5
```

Sample Output 0

```
1.118034
```

Question 8 – Candy Party

Jeff and Haif are siblings whose favourite game is a placement puzzle called the Candy Party. The game contains 81 cups arranged in a 9 x 9 grid, which is further divided into 3 x 3 boxes also called regions or blocks. Meanwhile, the game also contains 81 candies labelled 1 to 9. The objective of the game is to fill the empty cups with the candies so that every row, every column and every 3 x 3 box contains the candies labelled 1 to 9.

Your task is to help Jeff and Haif to validate whether their arrangement of candies conforms to the rule of the game.

Input Format

The first line and the subsequent 8 lines, represents the row, *i* of the Candy Party grid which contains nine positive integers, each separated by a space, which represents the column, *j* of the Candy Party grid..

Refer the sample input for illustration

Constraints

- **The row, i**: A positive integer where (1 <= i <= 9).
- **The column, j**: A positive integer where (1 <= j <= 9).

Output Format

Output “Valid” if the candy arrangement is valid or “Invalid” if the candy arrangement is invalid, without the double quotes.

Refer the sample output for illustration.

Sample Input 0

```
4 2 3 6 1 8 7 9 5
1 7 5 4 9 2 6 8 3
9 8 6 5 3 7 1 4 2
5 6 8 2 7 4 9 3 1
7 1 4 9 5 3 8 2 6
2 3 9 1 8 6 4 5 7
3 9 2 7 4 1 5 6 8
8 5 7 3 6 9 2 1 4
6 4 1 8 2 5 3 7 9
```

Sample Output 0

```
Valid
```

Question 9 – Venue Booker

Amir is a meeting room manager in the company that he works in. The staffs in the company resorts to Amir to book for the meeting room. In a given day, Amir has *N*number of bookings for meetings with their start time and finish time. Amir's job is to allocate the meeting room for as many meeting bookings assuming the room can only accomodate a single meeting at a given time.

Note: The start time and end time of any two activities may coincide.

Input Format

The first line contains an integer *N*, which is the number of bookings for the meeting room. The second line contains *N* integers separated by a space, which are the starting time of each booked meeting. The third line contains *N* integers separated by a space which are the finishing time of each booked meeting.

Constraints

- **The number of bookings, N:** A positive integer where (1 <= N <= 100) .
- **The starting time of the i-th booking, S:** A positive integer where (0 <= Si <= 1000).
- **The finishing time of the i-th booking, F:** A positive integer where (0 < Fi <= 1000) and Si < Fi

Output Format

Output the maximum number of meetings that could be allocated in the meeting room.

Sample Input 0

```
6
1 3 0 5 8 5
2 4 6 7 9 9
```

Sample Output 0

```
4
```

Question 10 – A Wise Leader

You are the head of a department in your company and your job is to delegate tasks to your co-workers. You have N number of co-workers under your supervision and N number of tasks for each of those co-workers. Each co-worker has his own capacity in terms of the time he takes to complete the task. Each co-worker is assigned with only one task and each task has to be done by exactly one person.

Your task is to assign the tasks to your co-workers in such a way that the total time taken to complete the set of tasks is the minimum.

Input Format

The first line contains an integer N which represents the number of tasks and the number of co-workers under your supervision. The following line contains N x N positive integers. The first N of these integers represents the time taken by the first co-worker to complete the N tasks. The next N integers represent the time taken by the second co-worker to do the N jobs and so on till the N-th person.

Constraints

- The number of co-worker, N = number of tasks, N: A positive integer where (1 <= N <= 11).
- The time taken to complete a task, T : A positive integer where (0 <= T <= 1000).

Output Format

Output the total minimum time taken in the optimal task assignment.

Sample Input 0

```
2
3 4
5 10
```

Sample Output 0

```
9
```

Explanation 0

The first co-worker takes 3 units and 4 units of time to complete task 1 and 2 respectively. The second person takes 5 units and 10 unit of time to complete task 1 and 2 respectively. We can see that the optimal delegation of task will be to give task 2 to co-worker 1 and task 1 to co-worker 2 for a total of 4 + 5 = 9 units of time.